CLAIMS

What is claimed is:

1. A system, comprising:

a main stack;

a micro-stack coupled to the main stack;

a data flag coupled to the micro-stack;

a stack pointer;

wherein the micro-stack resides in the core of a processor and the main stack resides outside of the core of the processor;

wherein the stack pointer indicates the top of the main stack; and

wherein the data flag indicates valid data in the micro-stack.

2. The system of claim 1, further comprising a computing engine coupled to the micro-stack, wherein the computing engine executes stack-based instructions.

3. The system of claim 2, wherein the micro-stack provides the computing engine with an operand.

4. The system of claim 1, wherein data are written to the micro-stack and wherein data are written to the main stack when the micro-stack is flushed.

5. The system of claim 1, wherein data are written to the micro-stack and wherein data are written to the main stack during an overflow condition.

6. The system of claim 1, wherein the data flag indicates coherence between the main stack and the micro-stack.

7. The system of claim 6, wherein coherency is established by examining the data flag and updating the main stack with values from the micro-stack.

8. The system of claim 1, wherein the micro-stack transfers data to the main stack when the micro-stack is full.

9. The system of claim 1, wherein the micro-stack retrieves data from the main stack when the micro-stack is empty.

10. The system of claim 1, wherein the size of the micro-stack is optimized for increased performance.

11. A method of managing a stack-based system, comprising:

loading data on a micro-stack and a main stack, wherein the micro-stack resides in the core of a processor, and the main stack resides outside of the core of a processor;

associating a data flag with each data loaded in the micro-stack;

determining the status of the data in the micro-stack; and

providing data to a compute engine from either the main stack or the micro-stack depending on the status of the data in the micro-stack.

12. The method of claim 11, wherein the data flag indicates the validity of the data in the micro-stack.

13. The method of claim 12, wherein the data flag indicates that the data in the micro-stack is valid and the data provided to the compute engine comes from the micro-stack.

14. The method of claim 12, wherein the data flag indicates that the data in the micro-stack is invalid and the data provided to the compute engine comes from the main stack.

15. The method of claim 12, further comprising transferring data from the micro-stack to the main stack if valid data is going to be overwritten.

16. The method of claim 12, further comprising transferring data from the main stack to the micro-stack if requested data is invalid.

17. The method of claim 11, wherein the data flag includes a read pointer and a write pointer.

18. The method of claim 11, wherein the data flag includes valid bits.

19. The method of claim 11, further comprising removing data from the micro-stack and disabling the valid data flag associated with each data removed from the micro-stack.

20. The method of claim 11, wherein the size of the micro-stack is adapted to provide reduced power consumption.